NORTHWEST NAZARENE UNIVERSITY

A Modification to Meal Ticket's Global Trade Item Number Validation to Use GS1 Data

THESIS
Submitted to the Department of Mathematics and Computer Science
in partial fulfillment of the requirements
for the degree of
BACHELOR OF SCIENCE

Bryn A. Gautier
2023

THESIS
Submitted to the Department of Mathematics and Computer Science
in partial fulfillment of the requirements
for the degree of
BACHELOR OF SCIENCE

Bryn A. Gautier
2023

A Modification to Meal Ticket's Global Trade Item Number Validation to Use GS1 Data

Author: Bryn Gautier
Bryn Gautier

Approved: Barry Myers
Barry L. Myers, Ph.D., Department of Mathematics and Computer Science, Faculty Advisor

Approved: Nathan D. Bizeau
Nathan D. Bizeau, Second Reader

Approved: Barry Myers
Barry L. Myers, Ph.D., Chair, Department of Mathematics and Computer Science

Abstract

A Modification to Meal Ticket's Global Trade Item Number Validation to Use GS1 Data.
GAUTIER, BRYN (Department of Mathematics and Computer Science).

Distribution companies throughout the world use Global Trade Item Numbers (GTINs)--the number used to create barcodes--to identify products. Because of this, it is important that these numbers be accurate, but many food distribution companies do not verify that the GTINs on their products are correct. Meal Ticket uses the GTIN checksum to invalidate GTINs, but there are many GTINs which pass the checksum and are not correct. So, this project used information from GS1, the company which assigns the company prefixes to create GTINs, to assist in validating GTINs within Meal Ticket's database. GS1 has limiting factors for what forms are valid for a company prefix, so the project used that information to validate and invalidate GTINs. GTINs with company prefixes of an incorrect format are invalidated. GTINs with a correct format company prefix are checked against GS1's API to validate that the GTIN belongs to a registered company. If a GTIN has a valid form company prefix, but does not appear to belong to a registered company, it cannot be fully invalidated due to oddities in the GS1 system.

Acknowledgments

This project was undertaken as part of Meal Ticket, so all funding for the project was from them. Thanks to Meal Ticket for giving me the opportunity to be an intern on their data science team. Thanks to my dad, who showed me how Scala and Spark work. And thanks again to Dad for patiently showing me, over and over, how to do a git commit to get my code in BitBucket.

Table of Contents

Table of Figures

A Modification to Meal Ticket's Global Trade Item Number Validation to Use GS1 Data

1. Introduction

One of the primary methods of identifying products for manufacturers and distributors is to use Global Trade Item Numbers (GTINs). Meal Ticket is a company that writes software for food distributors. Some of Meal Ticket's processes are heavily reliant on GTINs, so the GTINs within Meal Ticket's database need to be correct for those processes to function accurately. Unfortunately, many GTINs which Meal Ticket receives from food distributors are not accurate. That is why Meal Ticket undertook this project.

2. Background

2.1 What is a GTIN?

GTINs are 14 digit numbers which are supposed to be unique to a product so that packagers can assign unique barcodes to items. The 14 digits are not random, but follow a particular pattern (see Figure 1). There are a few GTINs with either 8, 12, or 13 characters, but those are extremely rare in Meal Ticket's clientbase. Those few GTINs with lengths other than 14 digits are padded on the left side with zeroes up to 14 digits, as that produces a valid 14 digit GTIN from the shorter GTINs (GS1 US, 2021, p. 9). For the remainder of the document, unless otherwise specified, "GTIN" will refer to the 14 digit GTINs.

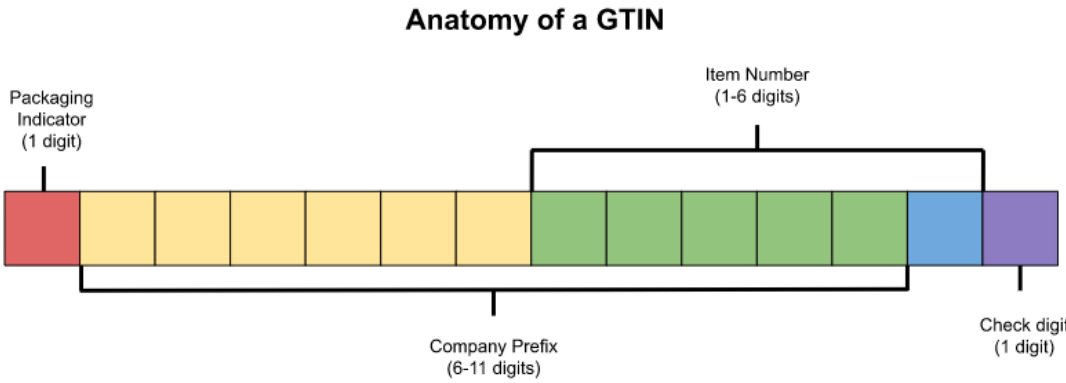

Figure 1. Anatomy of GTIN. Describes visually how a GTIN is formatted. Red: packaging indicator. Yellow: company prefix. Green: company prefix or item number. Blue: item number. Purple: check digit.

The last digit of a GTIN is a check digit (purple in Figure 1), which is calculated with a checksum formula based on the first 13 digits (see Figure 2). If the 14th digit does not match the calculated check digit, then the GTIN is incorrect. At the start of this project, calculating the check digit and comparing it to the 14th digit was the only known way to invalidate a GTIN.

Calculate Check Digit

ID Key Format	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}
Number **without** Check Digit	6	2	9	1	0	4	1	5	0	0	2	1	-
Step 1: Multiply	x	x	x	x	x	x	x	x	x	x	x	x	-
by	1	3	1	3	1	3	1	3	1	3	1	3	-
Step 2: Add results	=	=	=	=	=	=	=	=	=	=	=	=	-
to create **sum**	6	6	9	3	0	12	1	15	0	0	2	3	**=57**
Step 3: Subtract the **sum** from nearest equal or higher multiple of ten = 60- 57 = 3 (Check Digit)													
Number **with** Check Digit	6	2	9	1	0	4	1	5	0	0	2	1	**3**

Figure 2. Calculate Check Digit. How to calculate the check digit for a 13 digit GTIN. For a 14 digit GTIN, the only modification is that the number to multiply by in step 1 starts at 3 instead of 1. Image from GS1 (*How to calculate a check digit manually - services*, n.d.).

The first digit of a GTIN is a packaging indicator (red in Figure 1). This can be any digit, and is assigned by the company creating the GTIN. According to GS1, indicator digits are supposed to begin with 1-8 (GS1 US, 2021, p. 5), but in practice, brand owners do not follow this requirement. Over one third of the GTINs Meal Ticket receives from distributors begin with 0. Theoretically, the indicator digit is to be used to differentiate between products that are the same except for packaging level. For example, if a company sells chicken nuggets in two different sized packages, the GTINs for those packages should be the same except for the first digit, and therefore also the check digit.

Following the packaging indicator, a GTIN contains a company prefix (yellow and green in Figure 1), which must come from GS1, the company whose function is to assign company prefixes so that GTINs remain unique between companies. Each company that wishes to use GTINs must purchase a prefix from GS1. The company prefix is 6-11 digits in length. Any given company can receive multiple company prefixes from GS1, and can use any of its prefixes to create GTINs.

Besides the last digit, the remaining 1-6 digits belong to the item number (green and blue in Figure 1). The item number is assigned by the company creating a GTIN, and has no restrictions other than that it must contain only digits. Theoretically, the item number is to be used to distinguish among different types of products a company sells. In the above example with the chicken nuggets, the two packages of chicken nuggets should have the same item number. But if the company were to sell turkey nuggets, those should have a different item number because turkey nuggets are fundamentally different from chicken nuggets, no matter what size packaging is used.

The company creating a GTIN is only allowed to change the packaging indicator and the item number. The company prefix is assigned by GS1. The check digit is calculated based on the first 13 digits.

GTINs themselves are not assigned by GS1, only the company prefixes. It is up to whichever brand owner who purchased a company prefix to correctly create a GTIN for each product using that company prefix. According to GS1, "Brand owners are responsible for assigning GTINs to their products. The brand owner is the organization that owns the specifications of a trade item, regardless of where and by whom it is manufactured" (GS1 US, 2021, p. 9).

2.2 What is Wrong With Some of the GTINs in Meal Ticket's Database?

The majority of the GTINs Meal Ticket receives from distributors appear to be correct, though there is no way to know for certain that a GTIN is being used correctly. Some of the common problems among incorrect GTINs are described below.

GTINs commonly do not pass the checksum. These GTINs are easy to invalidate and Meal Ticket has been invalidating them for years, so identifying those was not part of this project.

GTINs that pass the checksum but are on thousands of products are most likely incorrect. There are duplicate products within Meal Ticket's system due to receiving the product either from multiple distributors or from the same distributor multiple times, so it is not guaranteed that a GTIN on multiple products is an incorrectly used GTIN. In fact, because there are duplicate products, there will be duplicate GTINs, and using those duplicate GTINs is the primary method for deduping the products. But if the same GTIN is on thousands of products with vastly different item names, then that GTIN is probably

being used incorrectly. In one instance, the same GTIN appeared on two products, one of which seemed to be blue cheese, and the other seemed to be a mouse trap. Blue cheese and mouse traps should never have the same GTIN. Spotting GTINs like this is a much more difficult task than finding GTINs which do not pass the checksum, as item names can be heavily truncated and different distributors might truncate them differently. The original goal of this project was to automate a way to find GTINs being used on different products, and mark those GTINs as invalid for the particular vendor which was using that GTIN incorrectly. Through the process of trying to find these GTINs, it was discovered that GTINs contain a company prefix, and the course of the project changed..

After it was discovered that GTINs contain a company prefix, more issues were found with the GTINs Meal Ticket received from distributors. First, company prefixes have limitations on what digits they can start with (see Appendix A). For example, they are not allowed to start with 2 (*GS1 Company Prefix*, n.d.). So, if the second digit of a GTIN is 2, that GTIN is incorrect because it contains a company prefix GS1 has reserved for internal use. Many GTINs within the system pass the checksum, but have an incorrectly formatted company prefix.

An additional issue with the GTINs Meal Ticket receives from distributors is that some of them have a prefix which has not been assigned by GS1 at all, but which appears to be correct in every other way (i.e. passes the checksum and has a correctly formatted prefix).

Again, many of the GTINs which Meal Ticket receives have no apparent problems. But the GTINs which do have problems can interfere with the accuracy of the

product deduper, which has many other processes built on top of it, so ideally every GTIN in the system should either be correct or not be used.

3. Methods

3.1 Invalid Form Company Prefixes

After it was discovered that company prefixes have restrictions on which numbers they can start with (see Appendix A), the course of the project changed to utilize this information.

Functions were either created or modified in three places within Meal Ticket's system: the development database, the system which processes incoming files from distributors, and part of the front end where clients can modify or enter product GTINs. In the development database, a new function was created called "gtin_is_valid" in addition to the "gtin_passes_checksum" function which already existed. This new gtin_is_valid function applies both the checksum and the restrictions on company prefixes. If the company prefix of a GTIN begins with any of the invalid beginnings or the GTIN does not pass the checksum, the function returns false. Otherwise, it returns true. Similar functions were created for the file processor and front end.

3.2 GS1 API

While work was being done on the company prefix validation functions, Meal Ticket gained access to GS1's Application Programming Interface (API), opening new directions for the GTIN project.

As mentioned in section 2.2, a GTIN can pass the checksum and have a company prefix which begins with valid digits, but still not be correct because the company prefix

is not actually assigned by GS1. Gaining access to GS1's API allowed Meal Ticket to verify that a GTIN had an assigned company prefix.

When running GTINs through GS1's API, the API either returns an empty string or a string of JavaScript Object Notation (JSON, a particular format of text for transferring or holding data). If the string was empty, the GTIN probably contained an unassigned company prefix. (See 3.2.1 for why it is only "probably.") If the string was JSON, then the GTIN had a valid company prefix, and the JSON would contain the information for the company which was assigned that prefix.

3.2.1 Problems With Using GS1's API

As the project delved deeper into GS1's API, several problems were encountered.

3.2.1.1 No Primary Key

First, GS1 has an account number for each company that acquires a company prefix. This allows GS1 to see if a company has multiple assigned prefixes. But the GS1 API does not return the account number, presumably for security purposes. This means that Meal Ticket does not have access to the primary key. This led those assigned the project to create artificial primary keys based on company location, entity global location number (which is similar to a GTIN) and other things. The exact details of the database table relationships will not be specified, as one does not need to understand the whole system to understand the GTIN part of the problem. Because Meal Ticket does not have the true primary key, it could have records from GS1 which appear to be from two separate companies, but which are really the same company, and that would never be discovered unless GS1's API began releasing a primary key.

But the problem of knowing if two records are really the same company has no bearing on whether or not a GTIN with one of those company prefixes is a valid GTIN. If the company prefix is in GS1's database, then it is valid no matter which company owns it.

3.2.1.2 Unable to Invalidate GTINs

Second, GS1's API returns the same value whether the company prefix of a GTIN is invalid or whether part of GS1's system is experiencing difficulties. This was discovered when GTINs which had previously been identified as invalid were run through GS1's API again in testing and they came back with company information. After troubleshooting, it was discovered that all of the GTINs which had been marked as invalid but later came back as valid belonged to GS1 Italy. The GS1 Italy server had been down when the GTINs were run through GS1's API the first time, and had been marked as invalid because they returned an empty string. Later, when the invalid GTINs were run through the API again later when GS1 Italy was back up, they were marked as valid because they returned JSON.

There is unfortunately no way to distinguish between GTINs which have an unassigned company prefix and GTINs with company prefixes that are stored in a part of the system that is currently down. This means that no GTIN can truly be invalidated by using GS1's API. Because of this, all GTINs marked as having invalid company prefixes are rerun through GS1's API once per month, to hopefully find any GTINs which were incorrectly marked as invalid.

4. Conclusion

The original purpose of the project, to identify GTINs which were valid but being used incorrectly and mark them as bad for the particular company misusing them, was not accomplished. Instead, a new method of invalidating GTINs and a new method of validating them were discovered.

First, the system can now recognize when a GTIN contains a company prefix that starts with invalid digits. This is the new way to invalidate GTINs. Second, GTINs which pass the checksum and have correctly formatted company prefixes can be run through GS1's API to determine if the company prefix is assigned by GS1. Due to the way GS1's API returns results of validating company prefixes GTINs which return as having an unassigned prefix may actually be from a part of GS1's system experiencing difficulties. When GTINs return company information, this is the new way of validating GTINs. Figure 3 shows a decision tree for determining whether GTINs are valid or not, based on Meal Ticket's system.

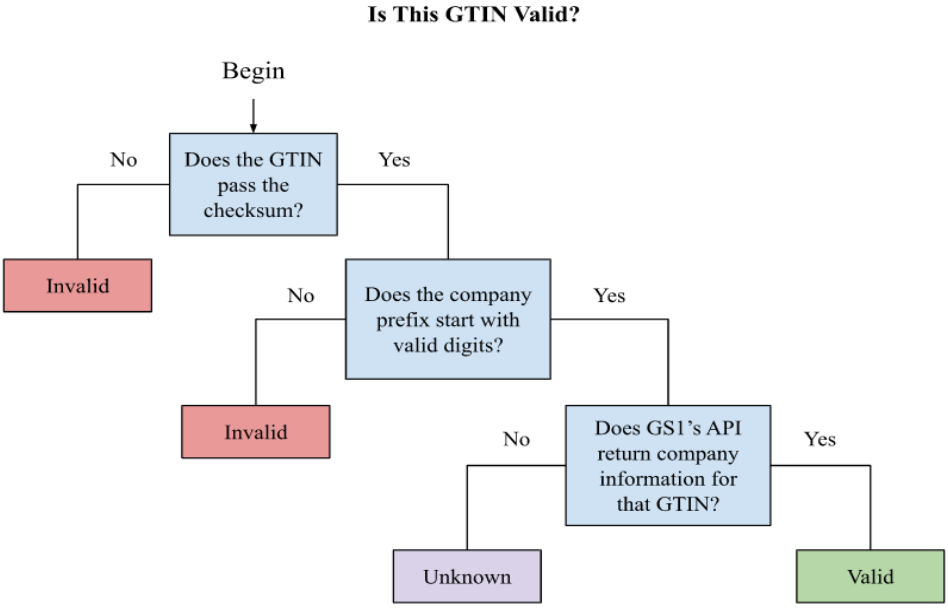

Figure 3. Is This GTIN Valid? Decision tree/flow chart for determining validity of a GTIN.

4.1 What I learned

This project taught me many new things about computer science, and data science in particular. The project primarily used Scala, a modern functional programming language. Because of this I learned a programming method completely different from object-oriented programming. The programming for this project was not pure functional programming (functions with no side effects), but it was still functional in design.

The project also used Apache Spark, so I got to use cluster computing. That was of particular interest to me, since large-scale data analysis is a significant part of data science. I discovered that I enjoy working with millions of rows of data and finding patterns and anomalies in the data, then fixing or filtering out the anomalies, or ignoring them when that is the most preferable option.

Part of using Scala and Spark together meant I also used Spark-SQL. It is very similar to the SQL variants most developers are familiar with. One difference is that it is used to access Scala DataFrames, not database tables. But Scala DataFrames are very similar to database tables in function and appearance while a program is running, though they are not used for permanent data storage. Likely the most significant difference at a glance is that the syntax of Spark-SQL is different from plain SQL, though the logic and key words are the same.

4.2 Future Work

The original purpose of the project was not accomplished. Rather new ways were found to validate or invalidate GTINs, but that does not help identify GTINs which are valid but being used incorrectly. For example, using the same GTIN on different

products. Finding a way to automatically identify incorrectly-used but valid GTINs would be a valuable future project.

If GS1 updates its API such that it is possible to distinguish between GTINs with unassigned prefixes and GTINs which were run through the system while it was down, then it will be possible to completely validate or invalidate GTINs. If such an update ever occurs, then Meal Ticket's system will need to be updated such that all GTINs which come back as invalid are marked as invalid and not rerun again.

References

GS1 US. (2021, July). *An introduction to the global trade item number (GTIN) - GS1 US*. Retrieved March 31, 2023, from https://www.gs1us.org/content/dam/gs1us/documents/industries-insights/standards/Guideline-An-Introduction-to-the-Global-Trade-Item-Number-GTIN.pdf

GS1. (n.d.). *GS1 company prefix*. Retrieved March 31, 2023, from https://www.gs1.org/standards/id-keys/company-prefix

GS1. (n.d.). *How to calculate a check digit manually - services*. GS1. Retrieved April 2, 2023, from https://www.gs1.org/services/how-calculate-check-digit-manually

Appendices

Appendix A

A list of the invalid beginnings for a company prefix, according to GS1, and what the numbers are reserved for (*GS1 Company Prefix,* n.d.).

Reserved Company Prefix Beginning	Reason for Being Reserved
0000000	Used to issue Restricted Circulation Numbers within a company
0000001 - 0000099	Unused to avoid collision with GTIN-8
020 - 029	Used to issue Restricted Circulation Numbers within a geographic region (MO defined)
040 - 049	Used to issue GS1 Restricted Circulation Numbers within a company
200 - 299	Used to issue GS1 Restricted Circulation Numbers within a geographic region (MO defined)
605, 606, 610, 614, 758, 894	Managed by GS1 Global Office for future MO
951	Global Office - *General Manager Number*
952	Used for demonstrations and examples of the GS1 system
977	Serial publications (ISSN)
980	Refund receipts
981 - 983	GS1 coupon identification for common currency areas
99	GS1 coupon identification